AMENDMENTS

In the claims:

Please replace the claims with the following listing of claims.

1. (Currently amended) An apparatus for updating a code image, comprising:

a loader configured to load a new code image into a temporary memory location separate from a memory space occupied by and used by an old code image;

a ~~conversion module~~ bootstrap module, within the new code image, configured to selectively reconcile incompatibilities between the old code image and the new code image; and

a copy module configured to copy the new code image into the memory space occupied by the old code image.

2. (Original) The apparatus of claim 1, wherein the old code image is updated substantially concurrent with normal execution of transactions by the apparatus.

3. (Original) The apparatus of claim 1, further comprising an initialization module configured to initiate execution of a run-time segment of the new code image.

4. (Original) The apparatus of claim 1, wherein the copy module copies the new code image into the memory space in response to reconciliation of the incompatibilities.

5. (Original) The apparatus of claim 1, further comprising a logic module configured to access version information for the old code image and version information for the new code

image and identify an incompatibility based at least in part on a difference between the version information.

6. (Currently amended) The apparatus of claim 5, wherein the ~~conversion~~ bootstrap module is configured to update modules that interface with the new code image based at least in part on a difference between the version information.

7. (Currently amended) The apparatus of claim 1, wherein the ~~conversion~~ bootstrap module recognizes persistent data associated with the old code image and associates the persistent data with the new code image such that the persistent data is available in response to execution of the run-time segment of the new code image.

8. (Original) The apparatus of claim 1, wherein at least one of the incompatibilities comprises different initialization requirements.

9. (Original) The apparatus of claim 1, wherein at least one of the incompatibilities comprises a difference between data structures used by the old code image and data structures used by the new code image.

10. (Original) An apparatus for updating a code image, comprising:

an update module configured to load a new code image into a temporary memory location separate from a memory space occupied by and used by an old code image; and

a bootstrap module within the new code image that executes subsequent to the update module, the bootstrap module configured to selectively reconcile incompatibilities between the old code image and the new code image prior to copying the new code image into the memory space occupied by the old code image.

11. (Original) The apparatus of claim 10, wherein the bootstrap module comprises a conversion module configured to reconcile incompatibilities based on version information for the old code image and the new code image and a copy module configured to copy the new code image over the old code image in response to reconciliation of the incompatibilities.

12. (Original) The apparatus of claim 10, wherein at least one of the incompatibilities comprises a difference between data structures used by the old code image and data structures used by the new code image.

13. (Original) A system that overlays an old code image with a new code image with minimal interruption of operations being performed by execution of the old code image, the system comprising:

a memory comprising an old code image and a buffer configured to store a new code image;

a processor executing instructions of the old code image to perform one or more operations, the processor configured to execute instructions of the old code image and the new code image;

a data structure configured to store an old code image pointer and a new code image pointer;

wherein, in response to an interrupt, the processor begins executing bootstrap code within the new code image, the bootstrap code configured to reconcile incompatibilities between the old code image and the new code image.

14. (Original) The system of claim 13, wherein the bootstrap code overlays the new code image in memory with the old code image in response to reconciliation of the incompatibilities.

15. (Original) The system of claim 14, wherein in response to the interrupt, the processor executes an update module of the old code image that loads the new code image into the buffer.

16. (Original) The system of claim 15, wherein the update module stores the old code image pointer and the new code image pointer in the data structure.

17. (Original) The system of claim 16, wherein the update module reads a new code image header identified by the new code image pointer to determine the location of the bootstrap code within the new code image.

18. (Original) The system of claim 17, wherein the bootstrap code identifies incompatibilities by reviewing capability fields of the old code image.

19. (Original) The system of claim 18, wherein the bootstrap code identifies incompatibilities by comparing version information of the new code image with version information of the old code image.

20. (Currently amended) A method for updating a code image, comprising:

loading a new code image into a temporary memory location separate from a memory space occupied by and used by an old code image;

selectively reconciling incompatibilities between the old code image and the new code image using bootstrap code of the new code image;

copying the new code image into the memory space occupied by the old code image.

21. (Original) The method of claim 20, wherein the old code image is updated substantially concurrently with execution of regular computer operations.

22. (Original) The method of claim 20, further comprising initiating execution of a run-time segment of the new code image.

23. (Original) The method of claim 20, wherein the new code image is not copied into the memory space until incompatibilities are reconciled.

24. (Original) The method of claim 20, further comprising accessing capability information for the old code image and capability information for the new code image and identifying an incompatibility based at least in part on a difference between the capability information.

25. (Original) The method of claim 24, further comprising updating modules that interface with the new code image based at least in part on a difference between the capability information.

26. (Original) The method of claim 20, further comprising determining persistent data associated with the old code image and associating the persistent data with the new code image such that the persistent data is available in response to execution of a run-time segment of the new code image.

27. (Original) The method of claim 20, wherein at least one of the incompatibilities comprises different initialization requirements.

28. (Original) The method of claim 20, wherein at least one of the incompatibilities comprises a difference between data structures used by the old code image and data structures used by the new code image.

29. (Currently amended) An apparatus for updating a code image, the apparatus comprising:

means for loading a new code image into a temporary memory location separate from a memory space occupied by and used by an old code image;

means for selectively reconciling incompatibilities between the old code image and the new code image using bootstrap code of the new code image;

means for copying the new code image into the memory space occupied by the old code image.

30. (Currently amended) An article of manufacture comprising a program storage medium readable by a processor and embodying one or more instructions executable by a processor to perform a method for updating a code image, the method comprising:

loading a new code image into a temporary memory location separate from a memory space occupied by and used by an old code image;

selectively reconciling incompatibilities between the old code image and the new code image using bootstrap code of the new code image;

copying the new code image into the memory space occupied by the old code image.